A SMASHING CASE STUDY

Creative Techniques for Creative Design Challenges
ICONOGRAPHY
WordPress is a wonderfully powerful and versatile features giving it the flexibility range of users. However, if you are a professional and plugins, sometimes these features can and options that allow off-the-shelf them cases can sometimes also be used to understand themes built for a specific use case.
Not only should the image be a good candidate for SVG, but SVG should also be a good candidate for the image.
<img src="my-image-1x.png" srcset="my-image-2x.png 2x, my-image-3x.png 3x" alt="my image" />
<picture>
  <source type="image/svg+xml" srcset="path/to/image.svg"/>
  <img src="path/to/fallback.png" alt="Image description"/>
</picture>
Better SVG Fallback and Art Direction With The <picture> Element

Besides using an SVG as a background image in CSS, you can serve SVG foreground images in HTML using one of several embedding techniques, each of which has its advantages and use cases. Unless
WHICH EMBEDDING TECHNIQUE?!
IT DEPENDS ™
1. Is the icon just decoration? If yes, I embed as a background image in CSS. I might embed it inline in the CSS to avoid an extra HTTP request.

2. Does the icon need to have a smooth transition? Does it need to have it? If yes, then don’t use as a background image: inline instead.

3. Does the icon serve a function? Does it replace text indicating a specific action the user can take? If yes, inline + provide accessibility requirements.

4. What kind of fallback suits this image best? Is text fallback better? or can it be a raster image? If the contents of the SVG contains text that is essential to convey meaning, text fallback might be the best option. For example: infographics.

5. What’s best for performance?

6. If inlining is best, how does that affect the markup?

7. ...
Supporter Plan

<image xmlns:xlink="http://www.w3.org/1999/xlink" width="100%" height="100%" x="0" y="0" xlink:href="/images/plan-supporter-no-text.svg"/>
<svg width="318.029" height="267.921" viewBox="0 0 318.029 267.921" aria-describedby="title">
  <title>Supporter Plan</title>
  <image xmlns:xlink="http://www.w3.org/1999/xlink" width="100%" height="100%" x="0" y="0" xlink:href="/images/plan-supporter-no-text.svg"></image>
  <text x="173" y="50" font-size=".8em">SUPPORTER</text>
</svg>
svg
text {
  font-family: $mija;
  fill: #fff;
  font-weight: bold;
}

SVG icons placed inline with the text can be sized to scale up and down with the text using text relative units.
figcaption{
  color: $grey;
  font-size: .8em;
  font-style: italic;
  padding: .75em 0;
}

::before {
  content: url('data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdmciIHN0cm9rZS13aWR0aD0iMjAwIiBoZWlnaHQ9IjIwMCIgZmlsbD0iIzIiIgZmlsbC1ydWxlPSJ0b3AiIiB2aWV3PSJodHRwOi8vd3d3LnczLm9yZy8xOTk5L3 WyIiB3aWR0aD0iMjAwIiBoZWlnaHQ9IjIwMCIvPjxwYXRoIGQ9Ik00MjQgNiA2MiA2My0xNDoiLCsikjxwYXRoIGQ9Ik0xNSA3IDU5LDE0IiB3aWR0aD0iMjAwIiBoZWlnaHQ9IjIwMCIvPjwvZGVmcz48L3N2Zz4=');
  display: inline-block;
  height: 1em;
  width: 1em;
  margin-right: .6em;
}
An icon font icon can inherit the same text color, making it more flexible.

-Jane Doe
<svg width=".." height=".." viewBox="..">
  <path d=".." fill="currentColor"/>
</svg>

figcaption svg {
  color: inherit;
}
Styling SVG <use> Content with CSS

An in-depth article on how to style the contents of the SVG <use> element and overcome some challenges it brings.
TYPOGRAPHY
CONTROLLING RATE OF SCALE

calc() & Viewport Units
vh, vw, vmin, vmax
Viewport-percentage lengths are relative to the size of the initial containing block, or the viewport, which is the visible part of the document inside the browser. When the height or width of the viewport is changed, they should be scaled accordingly.
Scalable Heading
:root{
  font-size: calc(16px + 3vw);
}

It appears that by using `calc()` and `vw` we can get responsive typography that scales perfectly between specific pixel values within a specific viewport range.

This means you can have perfect smooth scaling between any 2 font sizes over any viewport range. The font will start scaling and stop scaling exactly where you want.

-Mike Riethmuller
font-size: calc(16px + (24 - 16) * (100vw - 400px) / (800 - 400));

max font size - min font size
max screen size - min screen size
Responsive Typography

It appears that by using calc() and vw we can get responsive typography that scales perfectly between specific pixel values within a specific viewport range.

The problem with the common approach to responsive typography is that it is jumpy and requires a lot of media queries.

Viewport units are fluid but lack precise control over font-size.

Typically you might use a table like this to work out the range of font sizes across different resolutions.

<table>
<thead>
<tr>
<th>Viewport units:</th>
<th>1vw</th>
<th>2vw</th>
<th>3vw</th>
<th>4vw</th>
<th>5vw</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewport size</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>400px</td>
<td>4px</td>
<td>8px</td>
<td>12px</td>
<td>16px</td>
<td>20px</td>
</tr>
<tr>
<td>500px</td>
<td>5px</td>
<td>10px</td>
<td>15px</td>
<td>20px</td>
<td>25px</td>
</tr>
</tbody>
</table>
# Viewport units: vw, vh, vmin, vmax

Length units representing 1% of the viewport size for viewport width (vw), height (vh), the smaller of the two (vmin), or the larger of the two (vmax).

<table>
<thead>
<tr>
<th></th>
<th>IE</th>
<th>Edge</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Android Browser</th>
<th>Chrome for Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current aligned</td>
<td></td>
<td>Usage relative</td>
<td>Show all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>13</td>
<td>47</td>
<td>49</td>
<td>51</td>
<td>9.1</td>
<td>39</td>
<td>9.2</td>
<td>4.4</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>14</td>
<td>48</td>
<td>52</td>
<td>10</td>
<td>40</td>
<td>41</td>
<td>51</td>
<td>51</td>
<td></td>
</tr>
</tbody>
</table>

Notes: Known issues (11) Resources (7) Feedback

1. Partial support in IE9 refers to supporting "vm" instead of "vmin".
2. Partial support refers to not supporting the "vmax" unit.
3. Partial support in iOS7 is due to buggy behavior of the "vh" unit (see workarounds: 1, 2).
Truly Fluid Typography With vh And vw Units

By Michael Riethmuller

May 10th, 2016 | CSS, Responsive Web Design, Typography

32 Comments

Embracing fluid typography might be easier than you think. It has wide browser support, is simple to implement and can be achieved without losing control over many important aspects of design.
line-height: calc(1.3em + (1.5 - 1.3) * ((100vw - 21em)/(35 - 21)));
Flexible typography with CSS locks

In early 2012, I shared a formula for “molten leading”, or fluid line spacing. Experienced typographers know that long lines of text need more line spacing, but line spacing can be tighter for short lines of text. The problem is that on the web, our texts are flexible. So our line spacing needs to flex, too:

Manually adjusting line-height with media queries for optimum readability across vast number of screen sizes can be very hard. What makes it even harder, is, that instead of the screen width, the line-height should be relative to its container’s width and its font settings in order to achieve proper readability and appropriate spacing.
The math of CSS locks

September 2016

A CSS lock is a Responsive Web Design technique that lets you transition smoothly between two values, depending on the current viewport size, rather than jump straight from one value to the other.

This concept, and one implementation, were proposed by Tim Brown in Flexible typography with CSS locks.

When I tried wrapping my head around Tim's implementation, and creating variants of it, I had a hard time figuring out what was going on exactly. I did a lot of back-of-the-envelope calculations, and I thought it would be useful to share a mathematical explanation.

I'll describe the technique, its limitations, and the math that make it work. But don't worry about the math: it's basically addition and multiplication, and I've broken down the steps as much as possible—also there will be nice graphs.
TEXT UNDERLINES

CSS Gradients & Text Shadow
How to Roll Out New Features Without Hurting Loyal Users

by Vitaly Friedman
Crafting link underlines on Medium

How hard could it be to draw a horizontal line on the screen? It seems wrangling a few pixels together to stand in a file would be something computers should be pretty good at anno domini twenty-fourteen.

One would think so, but simple things are rarely simple under the surface… at least if they are worth anything. Typography, likewise, is a game of nuance. This is a story on how a quick evening project to fix the appearance of underlined Medium links turned into a month-long endeavour.

The history

Typography was never particularly fond of underlining. Do you want to emphasize your words? Grab some italics or use bold—hell, add some more...
So, the ideal technological solution would allow us to:

➤ change the width of the line (with additional half-pixel/retina support),
➤ change the distance from the text,
➤ change the color (even if just to simulate thinner width by using lighter grays instead of black),
➤ clear the descenders,
➤ (perhaps) have a separate style for visited links.
CSS Underlines Suck
Poster on: January 31, 2015

Rebuilding my personal website, I was reminded of just how bad text underlining support is in CSS. After reviewing countless blog posts, Stack Overflow answers, and CSS design articles, I’ve put together a (hopefully useful) survey of various hacks and workarounds that you can use to make underlining suck less.

Just want to know what to use? Skip to the bottom of this article.

Writing this post I was reminded of an popular earlier article written by Marcin Wichary of Medium, where he attempted to craft the "perfect" cross-browser underline. He wrote:

So, the ideal technological solution would allow us to:

- change the width of the line (with additional half-pixel/retina support),
- change the distance from the text,
- change the color (even if just to simulate thinner width by using lighter gray instead of black),
- clear the descenders,
- (perhaps) have a separate style for visited links.

If you haven’t read that article yet, do so now. I’ll wait.

Throughout this article, I’ll refer back to the Medium one, but this article covers more techniques, including libraries developed in response to Marcin’s post.

The W3C has a spec that solves some of this, but browser support is still practically non-existent. Firefox has support with a prefix, and Chrome has support, but only if you have the experimental features enabled.

To quote a developer from that thread,

"So to a user, it doesn’t work. An for a developer, using it is pointless, because no one will see it."
2.5. Text Decoration Line Continuity: the 'text-decoration-skip' property

<table>
<thead>
<tr>
<th>Name:</th>
<th>'text-decoration-skip'</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>none [ objects \ spaces \ ink \ edges \ box-decoration ]</td>
</tr>
<tr>
<td>Initial:</td>
<td>objects</td>
</tr>
<tr>
<td>Applies to:</td>
<td>all elements</td>
</tr>
<tr>
<td>Inherited:</td>
<td>yes</td>
</tr>
<tr>
<td>Percentages:</td>
<td>n/a</td>
</tr>
<tr>
<td>Media:</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value:</td>
<td>as specified</td>
</tr>
<tr>
<td>Canonical order:</td>
<td>per grammar</td>
</tr>
<tr>
<td>Animation type:</td>
<td>discrete</td>
</tr>
</tbody>
</table>

Note: Level 4 of this specification defines additional values ('leading-spaces' and 'trailing-spaces'), and uses them in the initial value of this property. At the time of writing, this is experimental and may still change, which is why it has not been incorporated into this level.

This property specifies what parts of the element's content any text decoration affecting the element must skip over. It controls all text decoration lines drawn by the element and also any text decoration lines drawn by its ancestors. Values have the following meanings:

'none'

Skip nothing; text-decoration is drawn for all text content and across atomic inline-level boxes.
Intriguing styles, eh?
color: $blue;
text-decoration: none;
padding: .5em 0;
background-image: linear-gradient($blue, $blue);
background-repeat: no-repeat;
background-size: 100% 2px;
background-position: center bottom 30%;

text-shadow:
3px 0 #fff,
2px 0 #fff,
1px 0 #fff,
-1px 0 #fff,
-2px 0 #fff,
-3px 0 #fff;
color: $blue;
text-decoration: none;
padding: .5em 0;
background-image: linear-gradient(currentColor, currentColor);
background-repeat: no-repeat;
background-size: 100% 2px;
background-position: center bottom 30%;
text-shadow:
  3px 0 #fff,
  2px 0 #fff,
  1px 0 #fff,
  -1px 0 #fff,
  -2px 0 #fff,
  -3px 0 #fff;
How to Roll Out New Features Without Hurting Loyal Users

by Vitaly Friedman

“Be agile, release early; release often.” We know the drill. But is it strategically wise to keep rolling out...
@mixin underline( $shadow-color: #fff, $border-color: currentColor, $border-width: 1px ) {
  text-decoration: none;
  padding: .5em 0;
  background: linear-gradient($border-color, $border-color) no-repeat; 
  background-size: 100% $border-width;
  background-position: center bottom 30%;

  text-shadow:
    3px 0 $shadow-color, 
    2px 0 $shadow-color, 
    1px 0 $shadow-color, 
    -1px 0 $shadow-color, 
    -2px 0 $shadow-color, 
    -3px 0 $shadow-color;

  transition: color .1s ease-out;
}
@mixin white_on_red_link ($shadow-color: $red) {
    color: #fff;
    @include underline($shadow-color, $border-color: #fff);
}

@mixin grey_on_navy_link ($shadow-color: $navy) {
    color: #9A97A9;
    @include underline($shadow-color, $border-color: #9A97A9);
}
Underline

Create beautiful underlines à la Medium! Now with the ability to clear descenders!

@include underline([$background, $color, $clear-descenders, $distance, $width])

Settings

- 'underline background': #f00
- 'underline color': #00e
- 'underline clear descenders': true
- 'underline distance': 1 (unitless number)
- 'underline width': 2 (unitless number)
- 'underline extend': false

Target

Creates specially formatted comments for use with gulp-css-target

@include target($target) { @content }

Set Multiple

Applies the same property to multiple values.

@include set-multiple($value, $property-list);
NESTED LINKS
“Be agile, release early; release often.” We know the drill. But is it strategically wise to keep rolling out new features often? Especially once a product you’re building reaches a certain size, you probably don’t want to risk the integrity of your application with every new minor release.
<a href="#">Parent</a>
<a href="#"></a>
Child
<a></a>
another child
</a>
<a href="#" target="_blank">Parent</a>

Child

another child

<a href="#" target="_blank">Parent</a>

Child

another child
<a href="#">Parent</a>
<object type="just/kidding">
  <a href="#">Child</a>
</object>
</a>
Another child
after wrapping any HTML with such attributeless `<object>` we would get just a wrapper element for this content. But a wrapper with an unusual trait: any content inside of it would be treated by browser’s parser without looking at the object’s context. So, using this trait we can, finally, nest one link into another, separating them for a parser.

-Roman Komarov
Nested Links

Read more...
How to Roll Out New Features Without Hurting Loyal Users

by Vitaly Friedman

“Be agile, release early; release often.” We know the drill. But is it strategically wise to keep rolling out new features often? Especially once a product you’re building reaches a certain size, you probably don’t want to risk the integrity of your application with every new minor release.
a {
    position: static;
    /* makes the heading link cover the entire article item area */
    &:before {
        content: "";
        display: block;
        position: absolute;
        top: 0;
        left: 0;
        width: 100%;
        height: 100%;
        z-index: 0;
        width: calc(100% + 80px);
        height: calc(100% + 60px);
        padding: 30px 40px 0 40px;
        left: -40px;
        top: -30px;
        transition: background-color .1s ease-out;
    }
}
background-color: transparent;
How to Roll Out New Features Without Hurting Loyal Users
by Vitaly Friedman

“Be agile, release early; release often.” We know the drill. But is it strategically wise to keep rolling out new features often? Especially once a product you're building reaches a certain size, you probably don't want to risk the integrity of your application with every new minor release.
THE PROBLEM

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fugiat suscipit ut odit animi consequatur error numquam perspiciatis voluptas quod eum.
This is one of those tricky CSS things that I see come up every few months. I guess what better place to address it than CSS-Tricks eh?

The situation involves ragged-right inline text. Like when a paragraph of text breaks to the next line whenever the next word won’t fit (i.e. most text on the internet). You want to add a background behind that text which:
box-decoration-break
### 5.4. Fragmented Borders and Backgrounds: the ‘box-decoration-break’ property

<table>
<thead>
<tr>
<th>Name</th>
<th>box-decoration-break</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>slice I clone</td>
</tr>
<tr>
<td>Initial</td>
<td>slice</td>
</tr>
<tr>
<td>Applies to</td>
<td>all elements</td>
</tr>
<tr>
<td>Inherited</td>
<td>no</td>
</tr>
<tr>
<td>Percentages</td>
<td>N/A</td>
</tr>
<tr>
<td>Media</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value</td>
<td>as specified</td>
</tr>
<tr>
<td>Animatable</td>
<td>no</td>
</tr>
</tbody>
</table>

When a break (page/column/region/line) splits a box, the ‘box-decoration-break’ property controls:

- whether the box’s margins, borders, padding, and other decorations wrap the broken edges of the box fragments
- how the background positioning area [CSS3BG] (and mask positioning area [CSS-MASKING-1], shape reference box [CSS-SHAPES-1], etc.) is derived from or duplicated across the box fragments and how the element’s background is drawn within them.

Values have the following meanings:

**clone**

Each box fragment is independently wrapped with the border, padding, and margin. The ‘border-radius’ and ‘border-image’ and ‘box-shadow’, if any, are applied to each fragment independently. The background is drawn independently in each fragment of the element. A no-repeat background image will thus be rendered once in each fragment of the element.

Note: Cloned margins are truncated on block-level boxes.
.featured-articles__teaser {

    span {
        background-color: #fff;
        padding: .3em .4em;
        line-height: 1.25em;
        box-decoration-break: clone;
    }
}
Don’t Miss These Articles on Smashing
z-index: 10;
CSS box-decoration-break - WD

Controls whether the box's margins, borders, padding, and other decorations wrap the broken edges of the box fragments (when the box is split by a break (page/column/region/line)).

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>prefixed</td>
<td>7.31% + 73.73% = 81.04%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7.31% + 4.88% = 12.19%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Browser</th>
<th>Current aligned</th>
<th>Usage relative</th>
<th>Show all</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE</td>
<td>8</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Edge</td>
<td>13</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Firefox</td>
<td>47</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>Chrome</td>
<td>49</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>Safari</td>
<td>9.1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Opera</td>
<td>39</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>iOS Safari</td>
<td>9.2</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>Opera Mini</td>
<td>51</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>Android Browser</td>
<td>4.4</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>Chrome for Android</td>
<td>4.4, 51</td>
<td>51</td>
<td></td>
</tr>
</tbody>
</table>

Notes: Partial support refers to working for inline elements but not across column or page breaks.
CUSTOM LIST
BULLETS

with CSS Counters
list-style-type
The list-style-type sets the appearance of list markers if the value of list-style-image—which can be used to set an image as a marker—is none or if the image it points to cannot be displayed.

A list marker can have one of three types: glyphs (circle, disc, square), numbering systems, and alphabetic systems.
Disc:
- First Item
- Second Item
- Third Item

decimal:
1. First Item
2. Second Item
3. Third Item

lower-latin (lower-alpha):
a. First Item
b. Second Item
c. Third Item
upper-roman:
I. First Item
II. Second Item
III. Third Item

Circle:
○ First Item
○ Second Item
○ Third Item

decimal-leading-zero:
01. First Item
02. Second Item
03. Third Item

upper-latin (upper-alpha):
A. First Item
B. Second Item
C. Third Item
georgian:
§. First Item
§. Second Item
§. Third Item

Square:
- First Item
- Second Item
- Third Item

lower-roman:
i. First Item
ii. Second Item
iii. Third Item

lower-greek:
a. First Item
β. Second Item
γ. Third Item

armenian:
Ա. First Item
Բ. Second Item
Գ. Third Item

none:
First Item
Second Item
Third Item
UNORDERED LIST

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5

ORDERED LIST

01 The Search for the Holy Grail
02 The Search for the Holy Grail
03 The Search for the Holy Grail
04 The Search for the Holy Grail
05 This is a smashing link
CSS COUNTERS
Basically, it allows us to number a list of elements of any type, the same way ordered lists are numbered. Numbers are then incremented per element, and displayed as generated content using CSS pseudo-elements. Using the corresponding CSS property, you can initialize a counter(s), and then specify the increment value, which specifies by how much the counter should be incremented for every occurrence of the element.
When writing articles, blog posts, tutorials, magazine entries or anything else, you will often want to include some images, charts, photographs, or even videos and code snippets to illustrate your content.

That being said, you will most likely want to attach some kind of caption to these elements, and perhaps number them so your readers can keep track of your thoughts.

And that’s exactly what we are going to do in today’s tutorial: combining the usage of the `<figure>` element with CSS counters to make your inserted elements (especially images) sexy as hell!
Automatically numbering headings via CSS

This post shows you how to number HTML headings with CSS. That is, given the following HTML:

```
<h1>My Article</h1>
<h2>Introduction</h2>
<h3>Rationale</h3>
<h2>Background</h2>
```

With the proper CSS, the above will be displayed as

```
My Article
1. Introduction
1.1. Rationale
2. Background
```
ol {
  list-style: none;
  counter-reset: listCounter; // Custom bullets: Decimal Leading Zero... but without the dot.
}

li {

  padding-left: 1.75em;

  counter-increment: listCounter; // for the custom bullets
  margin-bottom: 1em; // for the custom bullets

  /* Generate the list bullet */
  ::before {
    content: counter(listCounter, decimal-leading-zero);
    font-size: 0.5em;
    font-weight: bold;
    color: $red;
    font-family: Mija, Helvetica, Arial, sans-serif;
    margin-left: -2.5em;
    float: left;
  }
}
}
CSS Counters provide us with a way to apply automatic numbering using CSS.

Basically, it allows us to number a list of elements of any type, the same way ordered lists are numbered. Numbers are then incremented per element, and displayed as generated content using CSS pseudo-elements. Using the corresponding CSS property, you can initialize a counter(s), and then specify the increment value, which specifies by how much the counter should be incremented for every occurrence of the element.

Counters can be useful for numbering sections in articles, for example, or figures (images), and for tables of
Using CSS counters

CSS counters are, in essence, variables maintained by CSS whose values may be incremented by CSS rules to track how many times they're used. This lets you adjust the appearance of content based on its placement in the document. CSS counters are an implementation of Automatic counters and numbering in CSS 2.1.

The value of a counter is manipulated through the use of `counter-reset`, `counter-increment` can be displayed on a page using the `counter()` or `counters()` function of the `content` property.

Using counters

To use a CSS counter, it must first be reset to a value (0 by default). To add the value of a counter to an element, use the `counter()` function. The following CSS adds to the beginning of each h3 element "Section <the value of the counter>".

```css
body {
  counter-reset: section;
}

h3::before {
  counter-increment: section;
  /* Increment the section counter */
}
```
1. The order of the elements in the DOM needs to be logical and cannot be changed to accommodate the visual layout.

2. The content of the article in the DOM cannot be interrupted by any elements (e.g. <div>s) for presentation-only purposes.
FIRST ATTEMPT

1. Float Section (1) to the right.

2. Float Section (2) to the right—different width.

3. **Don’t** float Section (3)—let it wrap naturally around (1) and (2).

4. Add upper padding or margin to Section (3) to push it down to the same level of the heading as per the design.

5. TBC…
Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from divisions of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in section 1.10.32. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Necessitatibus eum, earum asperiores repellat minima dolore quos nemo doloribus modi, quae eveniet atque obcaecati praesentium, ut repellendus enim soluta sapiente amet!
FIRST ATTEMPT

1. Float Section (1) to the right.

2. Float Section (2) to the right—different width.

3. **Don’t** float Section (3)—let it wrap naturally around (1) and (2).

4. Add upper padding or margin to Section (3) to push it down to the same level of the heading as per the design.

5. Float the drop caps.
Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from divs 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in section 1.10.32. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Necessitatis eum, earum asperiores repellat minima dolorem quo nemo doloribus modi, quae eveniet atque obcaecati praesentium, ut repellendus enim soluta sapiente amet!

Sensible Automation

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from divs 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in section 1.10.32. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Necessitatis eum, earum asperiores repellat minima dolorem quo nemo doloribus modi, quae eveniet atque obcaecati praesentium, ut repellendus enim soluta sapiente amet!
OVERFLOW: HIDDEN;
Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from lines 10.32 and 10.33 of “de Finibus Bonorum et Malorum” (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in
Every time design needs to be changed because CSS is shit, a kitten dies somewhere.

-Lea Verou
HTML

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

CSS (SCSS)

```css
section-1 {  
  width: 100%;
  overflow: hidden;
  padding: 20px 0;
}

section-1__content {  
  background-color: red;
  width: 400px;
  height: 300px;
}
```

JS
1. Don’t float Section (1).
2. Float Section (2) to the right.
3. Float Section (3) to the left.
4. Pull Sections (2) and (3) up by applying a negative bottom margin to Section (1).
5. Push Section (2) back down using top margin.
6. Create pseudo-element inside Section (3) to create space where (3) overlaps with (1).
Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the origin, he discovered that "under the option..."
Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in section 1.10.32.

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a

Smashing Job Board

User Experience Designer (f/m) - blogfoster GmbH - (Berlin, Germany) - Fulltime
You will be part of the product management team that creates and shapes the products. Playing a central role in our team to create an enjoyable...

World-Class UI/UX Developer Needed! - Fortune Buildings, Inc. - (Work from Home) - Fulltime
We’re a fast-paced, fun real estate education company looking for a well-rounded LEAD UI/UX DEVELOPER for a 6-month project... potentially leading to full-time wo...

View more job openings
REQUIREMENTS

➤ Pull the component out of the article’s content flow.
➤ Make it fit into the “sidebar” width.
➤ Make the content that comes after the component flow where the component was flowing.
➤ Make component styles independent from parent context.
1. Pull the component to the right using negative margin.

2. Fit it to “sidebar” width by giving it a specific width.

3. Allow rest of the article content to wrap around it by floating it. This changes the way the content that comes after the component will behave around it.
STRETCH-OUT COMPONENTS
Be agile, release early, release often. We know the drill. But is it strategically wise to keep rolling out new features often? Especially once a product you’re building reaches a certain size, you probably don’t want to risk the integrity of your application with every new minor release. Read More

If you can’t get enough of this topic, our pal Yoav Weiss wrote an amazing chapter about the ins and outs of Responsive Images in Smashing Book 5. Get it today!

Add to Cart $39
width: 98vw;
position: relative;
left: 50%;
right: 50%;
margin-left: -49vw;
margin-right: -49vw;
width: 98vw;
position: relative;
left: 50%;
right: 50%;
margin-left: -49vw;
margin-right: -49vw;

-webkit-clip-path: inset(-50px 0 80px 0 round 11px);
clip-path: inset(-50px 0 80px 0 round 11px);
## CSS clip-path property

Method of defining the visible region of an element using SVG or a shape definition.

<table>
<thead>
<tr>
<th>Feature: CSS clip-path property</th>
<th>Global</th>
<th>Usage relative</th>
<th>Current aligned</th>
<th>Edge</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
<th>iOS Safari</th>
<th>Android Browser</th>
<th>Chrome for Android</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0% + 76.45% = 76.45%</td>
<td>0% + 7.71% = 7.71%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. Partial support refers to only supporting the `url()` syntax.
2. Partial support refers to supporting shapes and the `url(#foo)` syntax for inline SVG, but not shapes in external SVGs.
3. Supports shapes behind the `layout.css.clip-path-shapes.enabled` flag.
If you can't get enough of this topic, our pal Yoav Weiss wrote an amazing chapter about the ins and outs of Responsive Images in *Smashing Book 5*. Get it today!

[Add to Cart $39](#)
I bookmarked a tweet a few months ago, as it was a bonafide CSS trick and that’s what we love around here.

The issue is: how do we make a full-browser-width container when we’re inside a limited-width parent?
HEADS UP!
Viewport height is taller than the visible part of the document in some mobile browsers

18 février 2015
Web, Safari, iPhone, iPad, bug

When trying to use a 100vh CSS value to build a new interface for a game that would use the full viewport, I discovered that this full height value meant the bottom of the game interface was partially hidden behind the browser buttons bar or below the “fold” of some mobile browsers.

An issue with Apple iOS Safari

I first discovered this on my iPhone 5 and iPad 2.

Here is what this page looks like on an iPhone 5:
COMPONENT-LEVEL ENHANCEMENTS

with Flexbox
Web Image Effects Performance Showdown

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old.

Richard McClintock, a Latin professor, who was an expert on Latin, explained that the development of Lorem Ipsum has been the industry's standard model text ever since the 1500s, when it was popularised by desktop publishing software.

Sara Soueidan is a freelance front-end Web developer, author, and speaker from Lebanon. Lorem ipsum dolor sit amet. Sara is thoroughly enjoying building the Smashing Magazine pattern library. She absolutely loves Dan’s design and is loving turning it into a live Web site.
Web Image Effects Performance Showdown

ABOUT THE AUTHOR

Sara Soueidan is a freelance front-end Web developer, author, and speaker from Lebanon. Lorem ipsum dolor sit amet. Sara is thoroughly enjoying building the Smashing Magazine pattern library. She absolutely loves Dan’s design and is loving turning it into a live Web site.
Web Image Effects Performance Showdown

ABOUT THE AUTHOR

Sara Soueidan is a freelance front-end Web developer, author, and speaker from Lebanon. Lorem ipsum dolor sit amet. Sara is thoroughly enjoying building the Smashing Magazine pattern library. She absolutely loves Dan's design and is loving turning it into a live Web site.
Web Image Effects Performance Showdown

# hashtag 25  # CSS 13  # Responsive Images 61
# Performance 61

ABOUT THE AUTHOR

Sara Soueidan is a freelance front-end Web developer, author, and speaker from Lebanon. Lorem ipsum dolor sit amet. Sara is thoroughly enjoying building the Smashing Magazine pattern library. She absolutely loves Dan's design and is loving turning it into a live Web site.
“Be agile, release early; release often.” We know the drill. But is it strategically wise to keep rolling out new features often? Especially once a product you’re building reaches a certain size, you probably don’t want to risk the integrity of your application with every new minor release.
.article--grid__container {
  display: flex;
  flex-wrap: wrap;
  @include clearfix;
}

.article--grid {
  display: flex;
  flex-direction: column;
}
How to Roll Out New Features Without Hurting Loyal Users

by Vitaly Friedman

“Be agile, release early; release often.” We know the drill. But is it strategically wise to keep rolling out new features often? Especially once a product you’re building reaches a certain size, you probably don’t want to risk the integrity of your application with every new minor release.
How to Roll Out New Features Without Hurting Loyal Users
by Vitaly Friedman

“Be agile, release early; release often.” We know the drill. But is it strategically wise to keep rolling out new features often? Especially once a product you’re building reaches a certain size, you probably don’t want to risk the integrity of your application with every new minor release.
.col--article--grid {
  &:nth-of-type(1) {
    order: 1;
  }
  &:nth-of-type(2) {
    order: 2;
  }
  &:nth-of-type(3) {
    order: 3;
  }
  &:nth-of-type(4) {
    order: 0;
  }
  &:nth-of-type(n+5) { /* all items starting from the 5th one */
    order: 4;
  }
}
How to Roll Out New Features Without Hurting Loyal Users
by Vitaly Friedman

“Be agile, release early; release often.” We know the drill. But is it strategically wise to keep rolling out new features oftentimes? Especially once a product you’re building reaches a certain size, you probably don’t want to risk the integrity of your application with every new minor release.
/* TWO COLUMN GRID: Ad must be second in the list */

@media all and (min-width: 640px) {
    .col--article--grid {
        &\nth-of-type(1) {
            order: 1;
        }
        &\nth-of-type(2) {
            order: 3;
        }
        &\nth-of-type(3) {
            order: 4;
        }
        &\nth-of-type(n+5) {
            order: 5;
        }
        &\nth-of-type(4) {
            order: 2;
        }
    }
}
How to Roll Out New Features Without Hurting Loyal Users

by Vitaly Friedman

“Be agile, release early; release often.” We know the drill. But is it strategically wise to keep rolling out new features often? Especially once a product you’re building reaches a certain size, you probably don’t want to risk the integrity of your application with every new minor release.
/* THREE COLUMN GRID: Ad must be second in the list */

@media all and (min-width: 1100px) {
  .col--article--grid {
    &:nth-of-type(1) {
      order: 1;
    }
    &:nth-of-type(2) {
      order: 2;
    }
    &:nth-of-type(3) {
      order: 4;
    }
    &:nth-of-type(n+5) {
      order: 5;
    }
    &:nth-of-type(4) {
      order: 3;
    }
  }
}
@media all and (min-width: 1440px) {
  .col--article--grid {
    &:nth-of-type(1) {
      order: 1;
    }
    &:nth-of-type(2) {
      order: 2;
    }
    &:nth-of-type(3) {
      order: 3;
    }
    &:nth-of-type(n+5) {
      order: 5;
    }
    &:nth-of-type(4) {
      order: 4;
    }
  }
}
INTERACTIONS & ANIMATIONS
.SLIDEUP()

.SLIDEDOWN()
Don’t Miss These Articles on Smashing

Sara Soueidan wrote —— 2 hours ago

Breaking Out of the Box: Design Inspiration (August 2016) • 21 comments

Sara Soueidan wrote —— 2 hours ago

Breaking Out of the Box: Design Inspiration (August 2016) • 21 comments
.element {
  max-height: 0;
  transition: max-height .4s ease-in-out;
}

.element.open {
  max-height: 1000px; /* element would surely be less than 1000px */
}
.enhanced .slideDown {
  max-height: 50em;
  transition: all .9s ease-out;
}

&.main-nav {
  max-height: 50em;
}

&.search-loader-wrapper {
  max-height: 100px;
  transition-duration: .2s;
}

.enhanced .slideUp {
  max-height: 0;
  transition: all .3s ease-out;

  &.search-loader-wrapper {
    transition-duration: 0.1s;
  }
}
When you set a max-height value, the element will animate visually only as far as its real height, but the transition time will be divided over the entire length specified.
**EXAMPLE**

Computed element height: 250px;
max-height: 1000px;
transition-duration: 400ms

Visual result: transition to 250px over a course of: \((400 \times 250) / 1000 = 100\)ms.

That is, the transition is faster than you initially wanted it to be.

To get a 400ms transition for the 250px height, you’ll have to multiply the transition-duration by \(1000/250 = 4\).
.enhanced .slideDown {
  max-height: 50em;
  transition: all .9s ease-out;
}

&.main-nav {
  max-height: 50em;
}

&.search-loader-wrapper {
  max-height: 100px;
  transition-duration: .2s;
}

.enhanced .slideUp {
  max-height: 0;
  transition: all .3s ease-out;
}

&.search-loader-wrapper {
  transition-duration: 0.1s;
}
Responsive News

Where BBC News developers blog about responsive design.
Opinions expressed on this blog are those of the individual contributors, and are not necessarily those of the BBC as a whole.

Cutting the mustard

The browser is a hostile development environment and supporting a wide range of desktop browsers can be tough work.

One of the immediate challenges we discovered when we first started the responsive news prototype was the large range of devices that we would have to support. It terrified us. This article is about a solution we use to alleviate this problem.

Stats and market evolution
<script>

/* if JS is enabled and the browser is modern enough, add "enhanced" class name to
document (used for initially collapsing the navigation, for example) */
(function(){
  if(!('querySelector' in document
      && 'addEventListener' in window
      && 'getComputedStyle' in window)) {

    return;
  }

  window.document.documentElement.className += " enhanced";
})();

</script>
.main-nav__dropdown[target] {
    display: block; // opens when the menu link is clicked
}

.main-nav__list {
    display: block;
}
}
menu_button.addEventListener('click', function(e){
  e.preventDefault();
  this.classList.toggle('open');
  navigation_list.classList.add('slideUp');
})
PROVIDE TEXTUAL CONTEXT FOR ICONS
How to Roll Out Features With Loyal Users

by Vitaly Friedman

“Be agile, release early; roll the drill. But is it strategic rolling out new features a product you’re building you probably don’t want your application with ev
“Be agile, release early; roll the drill. But is it strategic rolling out new features to a product you’re building you probably don’t want your application with ev...
<div class="back-to-top-wrapper">
  <a href="#" class="back-to-top" title="Back to top">
    <svg viewBox="0 0 100 100" width="25px" height="25px" tabindex="-1" role="img" aria-labelledby="title">
      <title>Back to top</title>
      <image src="/images/back-to-top-arrow.png" xlink:href=""></image>
      <path d="M 10,50 L 60,100 L 70,90 L 30,50 L 70,10 L 60,0 Z" transform="translate(0 0) rotate(90 50 50)" fill="#fff" stroke="#fff"></path>
    </svg>
  </a>
</div>
<div class="back-to-top-wrapper">
  <a href="#" class="back-to-top" title="Back to top">
    <svg viewBox="0 0 100 100" width="25px" height="25px" tabindex="-1" role="img" aria-labelledby="title">
      <title>Back to top</title>
      <image src="/images/back-to-top-arrow.png" xlink:href=""></image>
      <path d="M 10,50 L 60,100 L 70,90 L 30,50 L 70,10 L 60,0 Z" transform="translate(0 0) rotate(90 50 50)" fill="#fff" stroke="#fff"></path>
    </svg>
  </a>
</div>
PROVIDE CLUES FOR CONTENT CHANGES
Responsive Images

A client-side solution for delivering alternate image data based on device capabilities to prevent wasted bandwidth and optimize display for both screen and print.
<input type="text" name="search" id="search-input" placeholder="Search Smashing..." aria-controls="search-results" / >
The aria-controls attribute creates a cause and effect relationship. It identifies the element(s) that are controlled by the current element, when that relationship isn’t represented in the DOM. For example a button that controls the display of information contained within a div.

[...]
The more widely the two elements are separated in the DOM, the more useful the aria-controls attribute becomes.

-Leonie Watson
When a User Agent (UA) supports aria-controls, it makes it possible for focus to be moved from the current element directly to the element it controls. The alternative is to navigate through all the intervening content in hopes of discovering what might have changed elsewhere on the page.

-Leonie Watson
A screen reader can only focus on one part of the page at a time. If something changes elsewhere on the page, there’s a good chance a screen reader user will be oblivious to it. When the update takes place within an ARIA live region, a screen reader is automatically notified (wherever its focus is at the time), and it conveys the updated content to the user.

-Leonie Watson
Interesting article covering how to offer great support with the minimum effort.

The core of automation is that machines are good at identifying patterns and humans are good at conveying emotions. An effective combination for both produces amazing results.
ARIA Live Regions

Introduction

In the past, a web page change could only be spoken in entirety which often annoyed a user, or by speaking very little to nothing, making some or all information inaccessible. Until recently, screen readers have not been able to improve this because no standardized markup existed to alert the screen reader to a change. ARIA live regions fill this gap and provide suggestions to screen readers regarding whether and how to interrupt users with a change.

Simple Live Regions

Dynamic content which updates without a page reload is generally either a region or a widget. Simple content changes which are not interactive should be marked as live regions. Below is a list of each related ARIA live region property with a description.

1. aria-live: The aria-live=POLITENESS_SETTING is used to set the priority with which screen reader should treat updates to live regions - the possible settings are: off/polite/assertive. The default setting is 'off'. This attribute is by far the most important.

2. aria-controls: The aria-controls=[IDLST] is used to associate a control with the regions that it controls. Regions are identified just like an ID in a div, and multiple regions can be associated with a control using a space, e.g. aria-controls="myRegionID1 myRegionsID2".

Not known if the aria-controls aspect of live regions is implemented in current ATs, or which. Needs research.

Normally, only aria-live="polite" is used. Any region which receives updates that are important for the user to receive, but not so rapid as to be annoying, should receive this attribute. The screen reader will speak changes whenever the user is idle.
PROVIDE CONTENT SHORTCUTS
The main content is not usually the first thing on a web page. Keyboard and screen reader users generally must navigate a long list of navigation links, sub-lists of links, corporate icons, site searches, and other elements before ever arriving at the main content. This is particularly difficult for users with some forms of motor disabilities.

-WebAIM
<a href="#main" class="skip-main">Skip to main content</a>
.skip-main {
  left: -999px;
  position: absolute;
  top: auto;
  width: 1px;
  height: 1px;
  overflow: hidden;
  z-index: -999;
}

.skip-main:active, .skip-main:focus {
  color: red;
  background-color: #fff;
  left: auto;
  top: auto;
  width: auto;
  height: auto;
  overflow: auto;
  padding: 0.5em 1em;
  text-align: center;
  font-size: 1.2em;
  z-index: 999;
}
"Skip Navigation" Links

Home > Articles > "Skip Navigation" Links

Article Contents
- Overview
- Creating "Skip Navigation" Links
- Browser Quirks
- Multiple "Skip" Links
- Alternatives to "Skip Navigation" Links

Overview
The main content is not usually the first thing on a web page. Keyboard and screen reader users generally must navigate a long list of navigation links, sub-lists of links, corporate icons, site searches, and other elements before ever arriving at the main content. This is particularly difficult for users with some forms of motor disabilities.
➤ Don’t treat accessibility as an afterthought.
➤ Test accessibility on a component level.
➤ The foundation of accessibility is proper markup. Write something you would be able to understand when CSS is disabled. (E.g. pricing table markup)
➤ Ask suitable questions before deciding which HTML elements to use.
➤ Enhance.
➤ Document. (Hacks, unconventional/uncommon tricks, etc.)
THANK YOU

https://sarasoueidan.com
https://twitter.com/SaraSoueidan